Source code: https://github.com/SavouryGin/computer_security_tasks

Story No.	CS-01	**Story Points**	
Title	Hide message using ending spaces		
User Story	**As** a sender, **I want** to hide some message in the file container, **so that** each bit of the message was encoded with a space at the end of the each line of the file.		
Scenario	1. The program requests a message from the sender. 2. The program requests the relative path to the container file from the sender. 3. The program turns the message into a string of bits. 4. The program opens a container file for reading. 5. The program creates a new file for recording information from the container file and from the bit string. 6. The program reads the file line by line. 7. If the bit of the secret message is 1, then the program appends one space at the end of the line of the file. 8. If the bit of the secret message is 0, then the program does not append anything. 9. The program saves the file with the secret message in the specified folder. 10. The program displays the message “Your message is hidden!”		
Result	File in the specified directory containing the secret message		

Story No.	CS-02	**Story Points**	
Title	Find message using ending spaces		
User Story	**As** a recipient, **I want** to find the message in the file container, **so that** information about bits of the secret message was obtained from spaces at the end of lines of the file.		
Scenario	1. The program asks the recipient if he wants to find the message. 2. If the user answers ‘yes’, then the program requests the path to the container file in which the message is hidden. 3. The program opens the specified file for reading. 4. The program reads the file line by line. 5. The program creates an empty string for recording secret message information. 6. If the last character of the string is a space, then the program writes 1 to the string. 7. If the last character of the message is not a space, then the program writes to string 0. 8. The program decrypts the bit representation of the message. 9. The program displays the secret message to the recipient.		
Result	Secret message displayed in the console		

Story No.	CS-03	**Story Points**	

Title	Hide message using double spaces
User Story	**As** a sender, **I want** to hide some message in the file container, **so that** each bit of the message was encoded with double spaces
Scenario	1. The program requests a message from the sender. 2. The program requests the relative path to the container file from the sender. 3. The program turns the message into a string of bits. 4. The program opens a container file for reading. 5. The program creates a new file for recording information from the container file and from the bit string. 6. The program reads the file line by line. 7. If the next bit of the secret message is 1, then in the text container the space doubles. 8. If the next bit of the message is 0, then the space in the text container remains one. 9. The program saves the file with the secret message in the specified folder. 10. The program displays the message “Your message is hidden!”
Result	File in the specified directory containing the secret message

Story No.	CS-04	**Story Points**	
Title	Find message using double spaces		
User Story	**As** a recipient, **I want** to find some message in the file container, **so that** information about bits of the secret message was obtained from double or single space.		
Scenario	1. The program asks the recipient if he wants to find the message. 2. If the user answers ‘yes’, then the program requests the path to the container file in which the message is hidden. 3. The program opens the specified file for reading. 4. The program reads the file line by line. 5. The program creates an empty string for recording secret message information. 6. The program ignores all characters except spaces. 7. If the space is single, then the program writes the character 0 to the string of the bit representation. 8. If the space is double, then the program writes the character 1. 9. The program decrypts the bit representation of the message. 10. The program displays the secret message to the recipient.		
Result	Secret message displayed in the console		

Story No.	CS-05	**Story Points**	
Title	Hide message using transliterated letters		
User Story	**As** a sender, **I want** to hide some message in the file container, **so that** each bit of the secret message was encoded using transliterated letters.		

Scenario	1. The program requests a message from the sender. 2. The program requests the relative path to the container file from the sender. 3. The program turns the message into a string of bits. 4. The program opens a container file for reading. This should be a text file containing Cyrillic text. 5. The program creates auxiliary arrays of information in which a one-to-one correspondence of Russian and English letters of a similar style is specified. These auxiliary arrays can be implemented as matching dictionaries. 6. The program creates a new file for recording information from the container file and from the bit string. 7. The program reads the file line by line. 8. The program ignores those characters that are not included in the matching dictionary. 9. If the next bit of the secret message is 1, then the program changes the Russian letter to an English letter with the same style according to the dictionary. 10. If the next bit of the secret message is 0, then the program leaves the Russian letter from the matching dictionary unchanged. 11. The program saves the file with the secret message in the specified folder. 12. The program displays the message “Your message is hidden!”
Result	File in the specified directory containing the secret message

Story No.	CS-06	**Story Points**	
Title	Find message using transliterated letters		
User Story	**As** a recipient, **I want** to find some message in the file container, **so that** information about bits of the secret message was obtained from transliterated letters.		
Scenario	1. The program asks the recipient if he wants to find the message. 2. If the user answers ‘yes’, then the program requests the path to the container file in which the message is hidden. 3. The program opens the specified file for reading. 4. The program reads the file line by line. 5. The program uses auxiliary arrays of information in which a one-to-one correspondence of Russian and English letters of a similar style is specified. 6. The program creates an empty string for recording secret message information. 7. The program ignores those characters that are not included in the matching dictionary. 8. If the next character from the matching dictionary is changed to English, then the program writes character 1 to the bit representation string. 9. If the next character from the matching dictionary is Cyrillic, then the program writes character 0. 10. The program decrypts the bit representation of the message. 11. The program displays the secret message to the recipient.		

Result	Secret message displayed in the console

Story No.	CS-07	**Story Points**	
Title	Search by signature of the file		
User Story	**As** a person responsible for computer security, **I want** to scan all the files in the specified folder **so that** all files that contain a specific signature were found.		
Scenario	1. The program asks for a signature from the user. 2. The program saves the first 16 bytes of the signature. 3. The program asks the user for the path to the folder to be scanned. 4. The program defines all subfolders and files of the specified folder and puts them into the list. 5. The program scans each file in the file list and looks for a sequence of bytes matching the signature. 6. The program saves the names and paths of all files containing this signature to the list. 7. The program displays a list of found files in the console.		
Result	A list of the paths of all files with this signature, displayed in the console.		

Story No.	CS-08	**Story Points**	
Title	Message encryption using the Vigenere method.		
User Story	**As** a sender, **I want** to encrypt a message in Russian using the Vigenere method **so that** only someone who knows the password can decrypt it.		
Scenario	1. The program asks the user for a message in Russian that needs to be encrypted. 2. The program asks the user for a password with which encryption will be performed. 3. The Russian alphabet is used for encryption (without the letter ‘ё’). 4. Each letter in the message is associated with a code. 5. Each letter in the password is also assigned a code. 6. Password letter codes are mapped to message letter codes. If the message is longer than the password, the password is repeated cyclically. 7. The code of each letter of the message changes in accordance with the Vigenere formula. 8. The program generates an encrypted message in accordance with the new letter codes. 9. The program displays an encrypted message on the screen.		
Result	The message encrypted using the Vigenere method, displayed on the screen.		

Story No.	CS-09	**Story Points**	
Title	Message decryption using the Vigenere method.		
User Story	**As** a recipient, **I want** to decrypt the message encrypted using the Vigenere method, **so that** I get a meaningful Russian phrase in accordance with my password**.**		

Scenario	1. The program asks the user for an encrypted message. 2. The program asks the user for a password. 3. The Russian alphabet is used for decryption (without the letter ‘ё’). 4. Each letter in the message is assigned with a code. 5. Each letter in the password is also assigned a code. 6. Password letter codes are mapped to message letter codes. If the message is longer than the password, the password is repeated cyclically. 7. The code of each letter of the message changes in accordance with the Vigenere formula. 8. The program generates a decrypted message in accordance with the new letter codes. 9. The program displays the decrypted message on the screen.
Result	The message decrypted using the Vigenere method, displayed on the screen.

Story No.	CS-10	**Story Points**	
Title	Folder encryption		
User Story	**As** a holder of secret data, **I want to** encrypt the data folder, **so that** only someone who knows the password can decrypt it.		
Scenario	1. The program asks the user for the path to the folder to be encrypted. 2. The program asks the user for a password for encryption. 3. The encrypts all files in a folder using the AES algorithm and the specified password. 4. The program archives the folder into a single encrypted file. 5. The program deletes the original folder with unencrypted files. 6. The program informs the user that the operation has been completed.		
Result	An encrypted file containing the source folder.		

Story No.	CS-11	**Story Points**	
Title	Folder decryption		
User Story	**As** a recipient of a file with secret data, **I want** to decrypt this file with my password, **so that** I get the original folder with the files to read.		
Scenario	1. The program asks the user for the path to the encrypted file. 2. The program asks the user for a password. 3. If the password is correct, the program unpacks all subfolders and files from the archive and decrypts them using the AES algorithm. 4. The program deletes the encrypted file. 5. The program informs the user that the operation has been completed.		
Result	The source folder with unencrypted files.		